

# Нестандартные нейросетевые архитектуры

Автор: Александр Вежнев [\\_vezhnick@mtu.ru](mailto:_vezhnick@mtu.ru)

Редактор: Владимир Вежнев [vvp@graphics.cs.msu.ru](mailto:vvp@graphics.cs.msu.ru)

*Нейронные сети применяются для решения задач классификации или кластеризации многомерных данных. Основная идея лежащая в основе нейронных сетей - это последовательное преобразование сигнала, параллельно работающими элементарными функциональными элементами.*

## Содержание

1. Введение.
2. Принцип «Победитель забирает всё».
  1. Статистические методы классификации, расстояние Махаланобиса.
  2. Сети с базисом радиального вида.
  3. Модель Хопфилда.
  4. Самоорганизующаяся сеть (карта) Кохенена.
  5. Гибридные нейросети.
3. Заключение.
4. Библиография.

## Введение.

Данная статья является продолжением статьи «Популярные нейросетевые архитектуры». Все архитектуры описанные в предыдущей статье [1], в той или иной степени, эксплуатируют идею персептрона. Однако, существуют нейросети принципиально иного характера. Далее разговор пойдёт именно о них. Ниже мы рассмотрим следующие архитектуры:

- Сети с базисом радиального вида и вероятностные сети.
- Модель Хопфилда.
- Самоорганизующиеся сети и сети с квантованием обучающих векторов.
- Гибридные сети

Перед чем перейти к рассмотрению конкретных сетей мы обсудим несколько общих идей.

## Принцип «Победитель забирает всё».

Для начала опишем одну важную концепцию - «победитель забирает всё» (Winner Takes All - WTA). Пусть перед нами стоит задача разбиения входных образов на  $n$  классов. Рассмотрим абстрактную сеть. Для этого предусмотрим в сети  $n$  выходов - каждый выход ассоциирован с конкретным классом. Тренировать сеть будем таким образом, чтобы на векторах определённого класса выход соответствующего нейрона был максимален, а на остальных нейронах выход был нулевым. Ясно, что на реальных данных результат будет отличаться от идеала. Концепция WTA состоит в следующем: после обработки очередного образа выберем из всех нейронов выходного слоя нейрон с максимальным выходом (нейрон победитель), далее повысим значение его выхода до максимума (обычно единица), а значение других нейронов выходного слоя положим равным нулю. Таким образом на выходе мы получаем вектор с единицей для соответствующего класса и с нулями для всех остальных. Подробнее об этой концепции можно прочитать в [1]. Применение данной концепции мы рассмотрим ниже.

## Статистические методы классификации, расстояние Махаланобиса.

Рассмотрим задачу классификации образов. Пусть есть  $k$   $n$ -мерных независимых случайных величин (моделей). Предположим, что они распределены нормально и  $\bar{\mu}_i$  - математическое ожидание, а  $\Sigma_i$  - ковариационная матрица соответствующей случайной величины (все величины -  $n$ -мерные случайные вектора). Из внешнего мира мы получили некоторый вектор  $\bar{x}$ , и нашей задачей является определить, к какой модели он относится (какая случайная величина из исходного набора его породила). Будем считать, что все модели равновероятны (нет априорных знаний). Как известно, из теории вероятностей, вероятность

$$p_i(x) = \frac{1}{(\sqrt{2\pi})^n \sqrt{|\det(\Sigma_i)|}} \exp\left[-\frac{1}{2}(\bar{\mu}_i - \bar{x}) \cdot \Sigma_i^{-1}(\bar{\mu}_i - \bar{x})^T\right]$$

принадлежности вектора  $\bar{x}$  к  $i$ ой модели

, обозначим  $A_i = \frac{1}{(\sqrt{2\pi})^n \sqrt{|\det(\Sigma_i)|}}$  (эта величина зависит только от модели и не зависит от  $\bar{x}$ ).

В показателе экспоненты стоит квадрат, так называемого, расстояния Махаланобиса. Формально расстояние Махаланобиса вводится следующим образом: пусть случайные величины  $\xi_1$  и  $\xi_2$  имеют одинаковую ковариационную матрицу  $\Sigma$ , тогда расстоянием по

Махаланобису назовём  $\|\xi_1 - \xi_2\|_M$ , квадрат которого вычисляется по формуле

$\|\xi_1 - \xi_2\|_M^2 = (\xi_1 - \xi_2) \cdot \Sigma^{-1}(\xi_1 - \xi_2)^T$ . Для задачи классификации (принятие решения о принадлежности вектора модели) вводится расстояние Махаланобиса между вектором  $\bar{x}$  и

моделью  $\xi$  с параметрами  $\bar{\mu}$  и  $\Sigma$   $\|\bar{x} - \bar{\mu}\|_M^2 = (\bar{x} - \bar{\mu}) \cdot \Sigma^{-1}(\bar{x} - \bar{\mu})^T$ . Что же обозначает расстояние Махаланобиса «физически»? Вычисление расстояния Махаланобиса можно рассматривать как вычисления длины вектора  $\bar{x}$  в некоторой системе координат связанной с математическим ожиданием модели. Рассмотрим подробно, как строится эта система координат. Первым шагом можно считать вычисление  $\bar{x}' = \bar{x} - \bar{\mu}$ . Это равносильно переносу системы координат из нуля в точку математического ожидания модели.

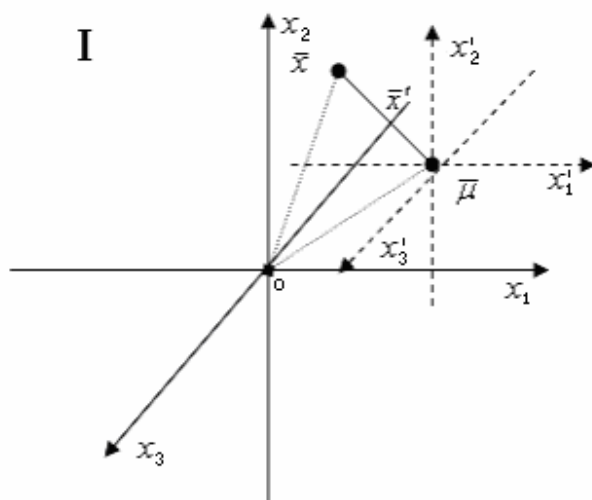


Рис1. Преобразование системы координат. Линейный сдвиг.

В новых обозначениях получаем  $\|\bar{x} - \bar{\mu}\|_{\Sigma} = \bar{x}' \Sigma^{-1} \bar{x}'^T$ , если бы ковариационная матрица была единичной (случай расстояния Евклида), то мы бы получили длину вектора  $\bar{x}$  в системе координат связанной с  $\bar{\mu}$ , то есть длину  $\bar{x}'$ . Рассмотрим случай, когда  $\Sigma$  является диагональной, это означает что элементы случайного вектора модели статистически независимы. В таком случае оси координат новой системы связанной с  $\bar{\mu}$ , как бы, растянутся пропорционально дисперсиям элементов. Длину вектора с началом в центре системы координат, можно воспринимать, как то во сколько раз нужно увеличить единичный пмерный шар, чтобы вектор лежал на его поверхности. Для наглядности рассмотрим, что произойдёт с единичным шаром в новой с.к. (системе координат).

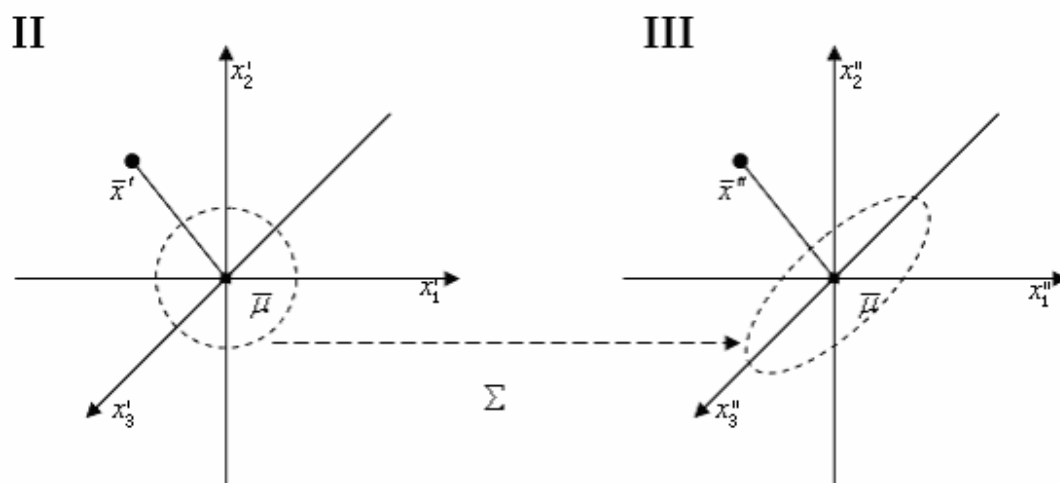


Рис2. Преобразование системы координат. «Растяжение».

Единичный шар в новой с.к. растянется по осям, пропорционально дисперсии, превратившись, с точки зрения предыдущих с.к., в эллипсоид. Таким образом длиной вектора нужно будет считать то, во сколько раз нужно увеличить эллипсоид (растянув его оси в равной пропорции), чтобы конец вектора лежал на его поверхности. Теперь вектора с одинаковой длиной образуют эллипсоид, а не сферу, с осями параллельными осям координат в системе связанной с  $\bar{A}$ . Таким образом изменение на одно и то же число двух различных координат входного вектора  $\bar{x}$ , скажется на его мере неравномерно, а пропорционально дисперсии по соответствующей координате. Теперь возьмём произвольную матрицу  $\Sigma$ . Как и в предыдущем случае единичный шар растянется превратившись в эллипсоид, вместе с этим недиагональные элементы  $\Sigma$  вызовут, как бы, вращение эллипсоида вокруг начала координат. Таким образом (из-за статистической зависимости координат) изменяя одну из координат вектора  $\bar{x}$  в исходной или второй с.к., мы изменяем сразу несколько в конечной с.к. (согласно их ковариации). Итак, основываясь на параметрах модели, мы получили новую с.к. в которой длина  $\bar{x}$  и есть расстояние Махаланобиса.

$$p_i(x) = A_i \exp\left(-\frac{1}{2} \|\bar{x} - \bar{x}_i\|_{\Sigma}^2\right)$$

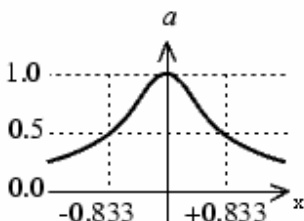
С учётом выше сказанного, можем записать . Таким образом, зная параметры моделей (мат. ожидание и ковариационную матрицу), для вынесения решения о принадлежности  $\bar{x}$  к той или иной модели (классу) достаточно посчитать вероятность для каждой из моделей  $p_i(\bar{x})$  и выбрать модель с максимальным значением, т.е.  $\bar{x} \in \bar{S}_i : p_i(\bar{x}) = \max_{j=1..k} (p_j(\bar{x}))$  - такой подход в выборе гипотез называется методом максимального правдоподобия.

Рассмотрим простейший случай: элементы случайных векторов (моделей) статистически независимы и имеют одинаковую дисперсию  $\sigma^2$ , следовательно диагональные элементы ковариационной матрицы равны между собой, а все остальные элементы - нули, т.е.

$\Sigma_i = \sigma^2 \cdot I$ , где  $I$  - единичная матрица. Тогда  $\|\bar{x}_1 - \bar{x}_2\|_{\Sigma}^2 = \frac{1}{\sigma^2} (\bar{x}_1 - \bar{x}_2)^T \cdot (\bar{x}_1 - \bar{x}_2)$ , так как произведение транспонированного вектора на самого себя есть квадрат евклидовой нормы, то  $\|\bar{x}_1 - \bar{x}_2\|_{\Sigma}^2 = \frac{1}{\sigma^2} \|\bar{x}_1 - \bar{x}_2\|_E^2$  (при  $\sigma^2 = 1$  мы получим простое расстояние в смысле Евклида).

### Сети с базисом радиального вида.

Данные нейросети тесно связаны с понятием функции радиального вида (Radial Based Functions - RBF)[5] и статистическими методами кластерного анализа. В контексте нейросетей под функцией радиального вида будем понимать простой Гауссиан:  $\varphi(x) = \exp(-bx^2)$ . Как видно из формулы  $0 < \varphi(x) \leq 1, \forall x \in R$ , притом своего максимума она достигает в нуле.



Выше приведён график при коэффициенте  $b=1$ . Изменяя  $b$  мы меняем «чувствительность» функции: то насколько быстро значение функции приближается к единицы, при стремлении аргумента к нулю. Перейдём к описанию самой сети.

**Модель нейрона** существенно отличается от модели нейрона архитектурах подобных персептрону. Итак у нейрона  $n$ -мерный вход  $\bar{x}$  и  $n$ -мерный вектор весов  $\bar{v}$ . Выход нейрона определяется по следующей формуле:  $\varphi(\|\bar{x} - \bar{v}\|)$ , где  $\varphi(p) = \exp(-b \cdot p^2)$ . То есть выходом нейрона является значение функции радиального вида от аргумента равного расстоянию (выбор меры для данного случая мы обсудим ниже) между входным вектором и вектором весов. Что же это даёт? Выход нейрона тем больше, чем ближе входной вектор к вектору весов. Если расстояние нулевое, то выход нейрона максимален (единица).

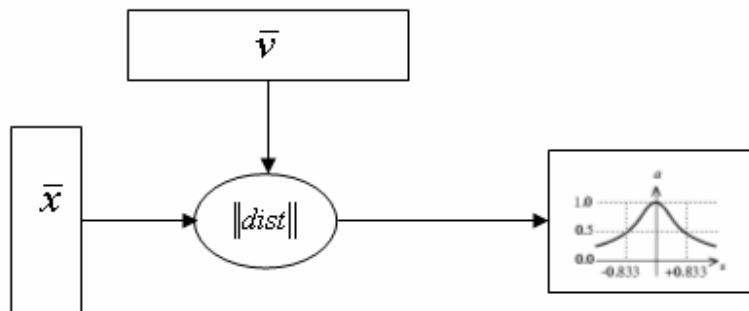


Рис3. Модель RBF нейрона.

Воспользуемся статистическим аппаратом классификации, описанным выше. Положим в нейроне вектором весов случайный вектор модели (о том, что в точности имеется ввиду мы

скажем ниже), т.е.  $\bar{v} = \bar{\xi}$ , свободный коэффициент  $b = \frac{1}{2}$ , а мерой положим расстояние по Махаланобису. Таким образом получаем, что выход нейрона - вероятность принадлежности входа к модели с точностью до коэффициента  $A$ . Как же понимать случайный вектор в качестве весов нейрона? Как видно из модели нейрона, веса применяются только для вычисления расстояния, следовательно от случайного вектора (модели) нам нужны только параметры -  $\bar{\mu}$  и  $\Sigma$ . В общем случае для произвольной модели нам потребуется хранить и то и другое. Иногда на практике (например, когда точные параметры модели не известны) элементы случайного вектора модели полагаются статистически независимыми и с одинаковой

дисперсией  $\sigma^2$  (последнего можно добиться искусственно, умножив каждый вход на  $\frac{\sigma}{\sigma_i}$ ), далее

$b = \frac{1}{2 \cdot \sigma^2}$  и  $\bar{v} = \bar{\mu}$ , а в качестве меры - евклидово расстояние. В итоге мы получим математически правильную модель не содержащую случайных векторов. Подчеркнём, что в отличие от персептрона, нейрон, в общем случае, характеризуется не одним вектором весов, а центром  $\bar{\mu}$  и матрицей  $\Sigma$ .

Рассмотрим возможные **структуры связей**. Рассмотрим сеть из одного слоя нейронов радиального типа.

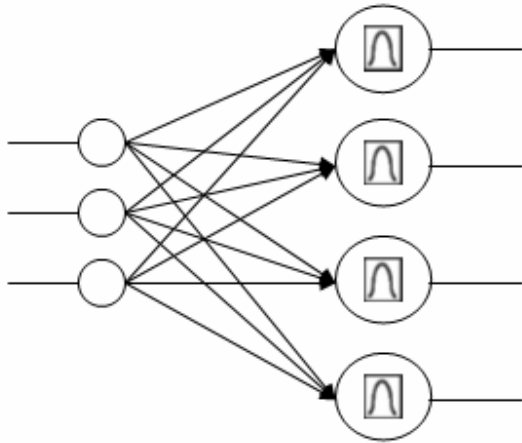


Рис4. Структура связей RBF сети.

В данном случае мы получим сеть, на выходе которой будут вероятности (с точностью до постоянного коэффициента) принадлежности входного вектора к моделям, определяемым весами нейронов из радиального слоя. Далее мы можем применить концепцию WTA, получив, таким образом, на выходе гипотезу о принадлежности вектора к модели. Такой вариант сети (с одним слоем радиальных нейронов и WTA) называется **вероятностной нейросетью** (обучение такой сети рассмотрим ниже). Понятно, что такая нейросеть реализует метод максимального правдоподобия.

Рассмотрим ещё один вариант. Пускай перед нами стоит не задача классификации, а задача моделирования отображения (как и в многослойном персептроне). Выстроим сеть следующей структуры:

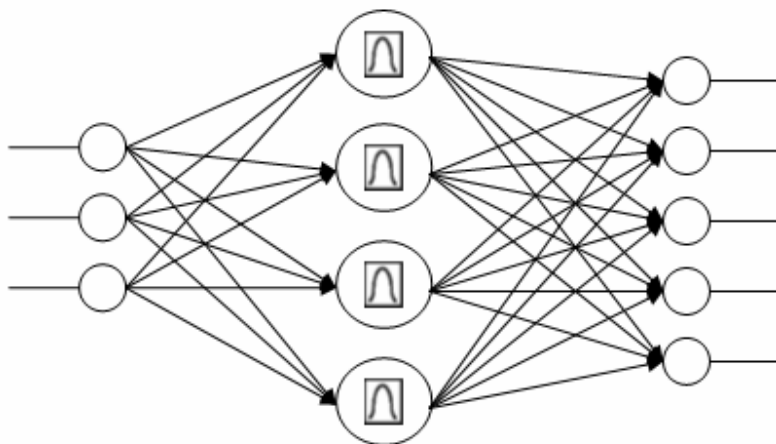


Рис5. Структура связей RBF сети с линейным слоем.

Как видно из картинке над слоем радиальных нейронов мы надстроили ещё один слой. В классическом варианте нейроны второго слоя обладают линейной функцией активации, то есть

$$y = \sum_i W_i \cdot x_i'$$

выход нейрона  $x_i'$ , где  $x_i'$  есть выход соответствующего нейрона радиального слоя, т.е.  $x_i' = \exp(-b \cdot (\bar{x} - \bar{\mu}_i) \Sigma_i^{-1} (\bar{x} - \bar{\mu}_i)^T)$ , где  $\bar{x}$  - входной образ сети. Таким образом мы получаем, что целевое отображение интерполируется с помощью линейной комбинации функций радиального вида - в данном случае n-мерных гауссианов. Функции радиального вида известны высоким качеством интерполяции и обладают лишь одним существенным недостатком - сглаживание или частичное поглощение локальных всплесков. Так же известны и архитектуры с более чем одним слоем нейронов с радиальной функцией активации [6], мы не будем подробно о них говорить, но утверждается, что нейросети такой архитектуры более эффективны. Ниже, в разделе «гибридные сети», мы ещё вернёмся к различным вариантам нейросетей радиального типа.

Рассмотрим методы **обучения** нейросетей с базисом радиального вида. Пусть нам дана обучающая выборка из пар  $(\bar{x}^a, \bar{y}^a), a=1, \dots, p$ , где  $\bar{x}^a$  - случайный вектор (модель) с параметрами  $\bar{\mu}_i^a$  и  $\Sigma_i^a$ , а  $\bar{y}^a$  - желаемый выход сети.

Начнём с рассмотрения вероятностной нейросети. Напомним, что под вероятностной нейросетью мы понимаем сеть с одним слоем радиальных нейронов и WTA на выходе. В данном случае  $\bar{y}^a$  будет вектором с одним элементом равным единице и остальными нулями. В таком случае, достаточно завести p радиальных нейронов, где каждый нейрон отвечает соответствующей модели и обладает её параметрами.

Таким же образом можно сконструировать и нейросеть с слоем линейных нейронов: Для каждой пары векторов из обучающей выборки заведём отдельный нейрон в радиальном слое с параметрами обучающего вектора. Таким образом каждый обучающий вектор формирует

$$y_i = \sum_j W_{ij} x_j$$

отдельный нейрон. Далее записав и решив простую систему  $y_i = \sum_j W_{ij} x_j$ , где  $y_i$  - соответствующий элемент  $\bar{y}^a$ ,  $W_{ij}$  - связь i-ого линейного нейрона и j-ого радиального, а  $x_j$  - выход соответствующего радиального нейрона.

На практике не всегда возможно узнать параметры и количество моделей заранее. Для их выявления применяются различные методы кластеризации и статистических оценок [7] (популярным методом является k-кластеризация). Для примера рассмотрим сеть с одним слоем радиальных нейронов и одним слоем линейных нейронов. Идея таких методов состоит в следующем:

- Пусть есть обучающая выборка  $(\bar{X}^a, \bar{Y}^a), a=1, \dots, p$ ,  $\bar{X}^a$  и  $\bar{Y}^a$  обычные числовые вектора.
- Разобьём пространство обучающих входных векторов на n кластеров.
- Для каждого кластера заведём по нейрону, с центром совпадающим с центром класса. Остальные параметры оценим с помощью статистических методов.

- Слой линейных нейронов настроим решив систему линейных уравнений, например методом наименьших квадратов (если нет аналитического решения).

Также есть метод постепенного конструирования сети с последовательным добавлением нейронов (используется в MatLab). Для обучения сетей с более чем одним слоем радиальных нейронов можно применить обратное распространение ошибки [6].

Нейросети с базисом радиального вида применяются для решения следующих задач:

- Распознавание образов.
- Интерполяция многомерных функций.
- Обработка и анализ звука.
- Анализ изображений.
- Обнаружение движения в видео последовательностях.
- Шумоподавление.

### Модель Хопфилда.

Среди нейросетей модель Хопфилда [2] занимает особое место. С помощью модели Хопфилда впервые удалось связать нейросети и нелинейные динамические системы и получить устойчивую модель ассоциативной памяти.

**Модель нейрона** довольно проста. Выход нейрона  $S$  может принимать два значения  $(-1, +1)$ .

Каждый вход нейрона  $x_i$ , как и в персептроне, обладает весом  $W_i$ . Обозначим  $h = \sum_i W_i x_i$

$$S(t+1) = \begin{cases} +1, h > \Theta \\ S(t), h = \Theta \\ -1, h < \Theta \end{cases}$$

Выход нейрона изменяется во времени по следующему закону  $S(t+1)$ , где  $\Theta$  - фиксированное число (обычно ноль).

**Структура связей** в модели Хопфилда радикально отличается от всего рассмотренного выше. В модели Хопфилда каждый нейрон связан со всеми остальными, то есть на вход нейрон получает сигналы от всех нейронов сети. Таким образом, нейросеть Хопфилда полностью характеризуется матрицей весов  $W$  ( $W_{ij}$  - вес связи  $i$ ого и  $j$ ого нейрона) и вектором состояния  $S$ , где  $S_i$  текущее значение выхода соответствующего нейрона. Обычно, связи считаются симметричными, то есть  $W_{ij} = W_{ji}$ . Так же предполагается что все диагональные элементы - нулевые  $W_{ii} = 0$ .

Рассмотрим поведение сети во времени при фиксированной матрице весов. Пускай в момент  $t=0$  мы передали вектору состояния некоторое значение. Существуют два варианта дальнейшего процесса работы сети: синхронный и асинхронный. Синхронный вариант работы заключается в том, что все нейроны одновременно изменят своё состояние в момент  $t+1$  согласно состоянию



сети на момент  $t$ . В случае асинхронной работы в момент  $t+1$  своё состояние изменяет только один нейрон, в момент  $t+2$  некоторый другой нейрон согласно состоянию сети в момент  $t+1$  и т.д. (каждый раз нейрон выбирается случайно). В любом случае с течением времени сеть каким-то образом изменяет своё состояние. Утверждается, что при наложенных нами условиях на матрицу весов сеть через какое-то конечное время придёт в стационарное состояние, то есть  $\exists T: \forall t \geq T, S(t) = S(t+1)$ . Так же утверждается, что это стационарное состояние  $S$  достигается и является одним и тем же вне зависимости от синхронности работы нейронов.

$$E(t) = -\frac{1}{2} \sum_i \sum_j S_i(t) W_{ij} S_j(t) + \sum_i S_i(t) \Theta_i$$

Рассмотрим выражение . Это выражение является, так называемой, функцией энергии сети. Эта функция является функцией Ляпунова для динамической системы, описываемой сетью Хопфилда. Можно доказать, что в процессе работы сети функция энергии будет не возрастать, а при стационарном состоянии - достигнет минимума. Таким образом сеть Хопфилда минимизирует соответствующий функционал энергии. Функционал с течением времени сходится к ближайшему от стартового состояния минимуму, т.е. если мы сможем умышленно создавать минимумы, то наша нейросеть сможет, как бы восстанавливать «повреждённые» входные образы. Поясним эту идею:

- Пускай мы смогли создать минимумы в состояниях  $S_1, S_2, \dots, S_n$  (достаточно далёких в смысле Хэмминга, то есть с большим числом различающихся элементов).
- Предположим, что у нас есть источник данных, который выдаёт двоичные вектора, являющиеся зашумлёнными или неполными вариантами  $S_1, S_2, \dots, S_n$  и мы хотим восстановить чистый вектор.
- Тогда получив из этого источника вектор  $\tilde{S}$ , введём сеть в состояние  $\tilde{S}$ .
- Дождёмся пока сеть придёт в стационарное состояние.
- Утверждается, что если шум не был слишком сильным, то сеть релаксируется к чистому вектору из  $S_1, S_2, \dots, S_n$ .

Отметим, что состояния сети являются вершинами  $n$ -мерного гиперкуба, а её динамика - траекторией, проходящей по его рёбрам. Таким образом, настраивая минимумы, мы как бы помечаем некоторые вершины, а в процессе работы сеть изменяется согласно кратчайшей траектории из начального состояния к какой-либо помеченной вершине. Такое свойство сети используется для решения, так называемой, задачи странствующего коммивояжера.

**Обучение** сети происходит по правилу Хебба: предположим мы хотим чтобы сеть запомнила бинарный вектор  $\bar{a}$ , тогда матрица весов вычисляется по следующему правилу  $W_{ij} = a_i a_j$ . Предположим мы уже обучили сеть некоторому количеству векторов, тогда для добавления ещё одного требуется пересчитать веса следующим образом:  $W_{ij}^{(p)} = W_{ij}^{(p-1)} + a_i^{(p)} \cdot a_j^{(p)}; W^0 = 0, p = 1, 2, \dots, k$ . Особой устойчивости сеть достигает при ортогональности обучающих образов. Утверждается, что сеть устойчиво хранит не более  $0.14N$  образов, где  $N$  число нейронов.

Полученная сеть является моделью с ассоциативной памятью, т.е. сеть хранит какой-то набор образов, а для доступа к одному из них требуется предъявить некоторый образ, «похожий» на требуемый. По такому же принципу работает и человеческая память: мы можем вспомнить объект, по его частичному или зашумлённому изображению. Противоположностью ассоциативной памяти является память адресная, где для получения некоторого хранящегося объекта нужно предъявить его уникальный номер - так работает память во всех современных компьютерах.

Отметим интересную особенность сети Хопфилда - ложная память. Действуя согласно правилу Хебба мы получим минимумы энергии в интересующих нас векторах, однако, так же мы получим минимумы и в некоторых точках, которым мы специально не обучали. Это сравнимо с эффектом «де жа вю» - нам кажется, что мы видели новый объект ранее.

Так же существуют различные модификации модели Хопфилда. Например, модель Коско с двунаправленной ассоциативной памятью [1] или модель с вероятностной нейродинамикой - машина Больцмана [2].

В машине Больцмана переход из одного состояния в другое осуществляется по следующему принципу:

$S_i(t+1) = \text{sign}(h_i(t) - \Theta_i)$ , с вероятностью  $P_i$ ,

$S_i(t+1) = -\text{sign}(h_i(t) - \Theta_i)$ , с вероятностью  $1 - P_i$ .

$$P_i = \frac{1}{1 + \exp\left(-\frac{(E_2 - E_1)}{T}\right)}$$

Где  $E_1$  - энергия начального состояния  $E_2$  - энергия конечного, а  $T$  - параметр. Обычно  $T$  называют температурой. При  $T \rightarrow 0$  - динамика сети близка к модели Хопфилда, при  $T \rightarrow \infty$  - абсолютно хаотична. Чем выше температура - тем больше вероятность перехода на уровень с большей энергией. На практике используется метод *имитации отжига* [2]: сеть начинает работу с относительно высокой температурой, но в процессе работы температура падает до нуля. Таким образом система тяготеет к более устойчивым минимумам, понижая, таким образом, количество релаксации к ложным (побочным) образам.

Нейросеть Хопфилда представляет большой научный интерес, однако, на практике применяется не часто; основная причина - работа только с бинарными векторами. Существуют, однако, и различные оригинальные математические решения, для преодоления этого ограничения для конкретных задач [9].

Сеть Хопфилда применяется в следующих областях:

- Минимизация функционалов.
- Задачи поиска (З.К.В).
- Физика.

- Цветовая сегментация [9].
- Распознавание образов.

### Самоорганизующаяся сеть (карта) Кохенена.

Выше мы рассматривали сети решающие задачи классификации при наличии некоторых априорных данных - обучающей выборки. В этом разделе мы рассмотрим нейросеть работающую на основе самоорганизации - сеть Кохенена (Kohonen Network, Self-Organizing Map, SOM) [2].

Сформулируем задачу многомерной кластеризации: пусть имеется некоторая многомерная выборка данных  $(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_7), \forall i: \dim(\vec{x}_i) = n$ , т.е. множество векторов некоторой фиксированной размерности, нам требуется построить инъективное отображение из множества входных данных в множество кластеров  $\Omega_1, \Omega_2, \dots, \Omega_k$ , где количество кластеров  $k$  обычно известно заранее, но ничего более о них не известно. Иными словами, требуется по некоторому признаку соотнести каждый вектор  $\vec{x}_i$  из входной выборки с каким-то кластером  $\Omega_i$ , тем самым разбив входное множество на  $k$  кластеров. Обычно вектора делят на кластеры по признаку близости к друг другу в смысле Евклида.

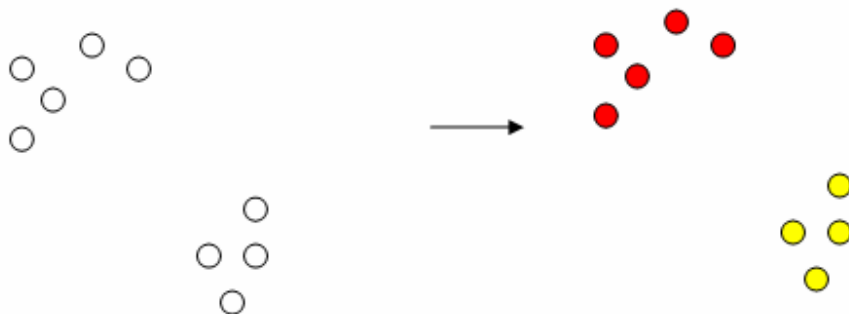


Рис6. Пример кластеризации.

На картинке приведён пример кластеризации двумерных векторов с числом кластеров  $k=2$ . Как видно, к одному кластеру мы отнесли те вектора, расстояние между которыми было невелико.

Итак начнём рассмотрение нейросети, решающую задачу кластеризации.

**Модель нейрона** чем-то напоминает нейрон сетей с базисом радиального вида, только с тождественной (линейной) функцией преобразования вместо Гауссиана.

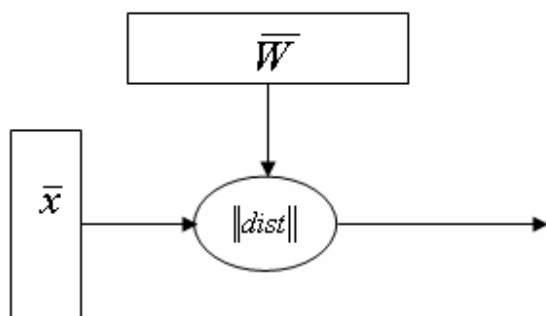


Рис7. Модель нейрона в Кохененской сети.

Результатом работы нейрона является евклидово расстояние между входным вектором  $\bar{x}$  и вектором весов  $\bar{W}$ .

Рассмотрим **структуру связей** сети Кохенена. Все нейроны сети находятся в одном слое и упорядочены в двумерную матрицу  $m \times n$ . Входной вектор поступает на входы каждому из нейронов, в следствие чего каждый нейрон производит определённый выход равный расстоянию входа до его вектора весов. Далее согласно WTA выбирается нейрон победитель.

**Обучение** в сети Кохенена происходит без учителя на основе самоорганизации. Пускай у нас имеется набор векторов  $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p\}, \forall i: \dim(\bar{x}_i) = n$ , для которого мы хотим провести кластеризацию. Проинициализируем веса нейронов небольшими случайными числами. Далее будем последовательно предъявлять сети вектора. Пускай на текущем шаге предъявив вектор сети мы получили нейрон победитель имеющий в матрице координату  $(i, j)$ . Подстроим веса нейронов следующим образом  $W_k(t+1) = W_k(t) + \eta(\bar{x} - W_k(t)) : \|(i, j) - (k, l)\| < \delta$ , где  $\bar{x}$  - входной вектор,  $\eta < 1$  - темп обучения, обычно убывающий со временем,  $\delta$  - размер окрестности который выбирается заранее. Таким образом мы оптимизируем веса нейронов из  $\delta$  окрестности нейрона победителя. Таким образом веса нейрона победителя и его соседей приближаются к входному вектору.

Проведя один или более сеансов обучения мы получим следующую картину - нейроны разделятся на группы отвечающие высоким выходом на определённые группы векторов из входного набора. Далее выбрав группы нейронов и приписав им определённый кластер будем относить вектор к тому кластеру, к которому относится нейрон победитель, если подать вектор на вход сети. Фактически мы спроецировали многомерное пространство входных векторов в двумерную матрицу.

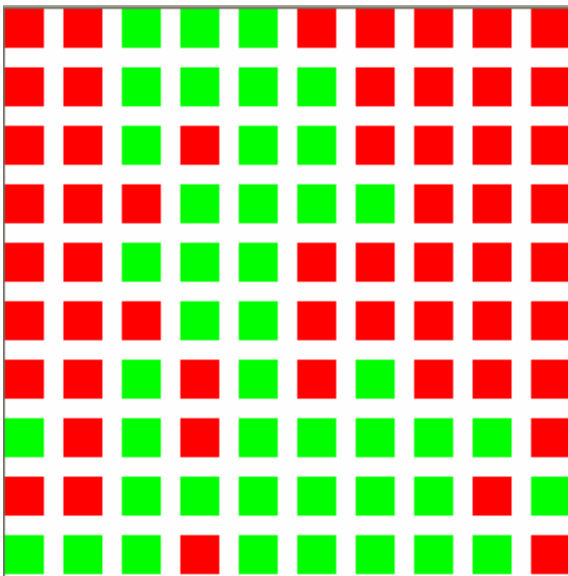


Рис7. Пример раскраски карты Кохенена.

На картинке показана карта нейронов в случае, когда мы хотим разбить входные векторы на 2 класса. Число кластеров может меняться от 2 до количества нейронов в сети.

Возможен и вариант обучения с учителем - так называемый алгоритм квантования обучающих векторов. Обучение по этому методу проходит следующим образом:

- Пускай нам заранее известно количество кластеров, притом существует обучающая выборка из множества  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p)$ ,  $\forall i: \dim(\bar{x}_i) = n$ , притом для каждого  $\bar{x}_i$  нам уже известен кластер  $\Omega_i$ .
- Вначале проведём несколько этапов обучения на основе самоорганизации, запоминая для каждого нейрона количество отнесенных к нему векторов того или иного класса.
- Разметим матрицу нейронов следующим образом - будем относить нейрон к тому кластеру, для которого максимально количество отнесённых к данному нейрону векторов.
- Далее будем проводить обучение со следующими изменениями - если нейрон победитель принадлежит тому же кластеру, что и обучающий вектор, то подстроим его веса и веса его соседей по прежней формуле:  $W_{kj}(t+1) = W_{kj}(t) + \eta(\bar{x} - W_{kj}(t)) : \|(i, j) - (k, l)\| < \delta$ , в ином случае по формуле:  $W_{kj}(t+1) = W_{kj}(t) - \eta(\bar{x} - W_{kj}(t)) : \|(i, j) - (k, l)\| < \delta$ .

Таким образом в случае правильной кластеризации веса нейрона и его соседей приблизятся к входному вектору, а в случае ошибки - удалятся.

К достоинствам сети можно отнести устойчивость, простоту, возможность анализа структуры кластеров. Недостатками являются: невысокая способность к обобщению, долгое время обучения.

Нейросеть Кохенена применяется в следующих задачах:

- Цветовая сегментация.
- Анализ звука.
- Анализ многомерных данных.

### Гибридные нейросети.

Как видно все описанные нейросети имеют свои недостатки и преимущества. Одним из способов преодоления недостатков какой-то конкретной архитектуры - гибридизация сетей. Имеется ввиду совмещение нескольких нейросетевых архитектур в одну, с целью повышения тех или иных качеств. Так же часто встречается гибридизация с не нейросетевыми моделями, то есть с другими методами распознавания образов. Известны гибриды нейросетей и скрытых Марковских моделей [10], систем нечёткой логикой [11], и т.д. Главным принципом при построение гибридной архитектуры, является использование той или иной модели в том, где она сильна и обход мест где она не срабатывает.

### Заключение.

Несмотря на видимый спад интереса к нейронауке, она продолжает жить и развиваться. В последнее время было сделано много продвижений в области интеграции нейронных сетей и систем с нечёткой логикой, построение гибридных сетей и т.д. За время своего существования нейросети пережили, как периоды небывалой популярности, так и моменты почти полного забвения. Следует понимать, что нейросети ни в коем случае не являются таким универсальным методом - они всего хороший и крайне удобный инструмент, со своими недостатками и преимуществами.

### Библиография:

[1] А.Вежневцев. Популярные нейросетевые архитектуры.  
[http://cgm.graphicon.ru/issue5/Paper\\_vap/index.html](http://cgm.graphicon.ru/issue5/Paper_vap/index.html)

[2] С. Терехов. Лекции по теории и приложениям нейронных сетей.  
 (http://alife.narod.ru/lectures/neural/Neu\_index.htm)

[3] J. L. Elman. Finding Structure in Time. Cognitive Science, 14, 179-211 (1990).

[4] H. Ninomiya, A Sasaki. A Study on Generalization Ability of 3-Layer Recurrent. Neural Networks. IJCNN '02.

[5] T. Koskela, M. Lehtokangas, J. Saarinen, K. Kaski. Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. <http://citeseer.ist.psu.edu/koskela96time.html>

[6] Е. Лисицин Radial Based Functions. Graphics and Media Journal.  
[http://cgm.graphicon.ru/issue2/radial\\_based/](http://cgm.graphicon.ru/issue2/radial_based/)

[7] J. Chao, M. Hoshino, T. Kitamura, T. Masuda. A multilayer RBF network and its supervised learning.

[8] M. Y. Mashor. Hybrid Training Algorithm for RBF Network. ICJIM'98  
[http://www.journal.au.edu/ijcim/may00/mashor\\_5.pdf](http://www.journal.au.edu/ijcim/may00/mashor_5.pdf)

[9] A.G. Bors, I. Pitas. Optical Flow Estimation and Moving Object Segmentation Based on Median Radial Basis Function Network. <http://citeseer.ist.psu.edu/312887.html>

[10] Da-chuan Cheng, Qin Pu, Kuo-sheng Cheng, Hans Burkhardt. Possibilistic Hopfield Neural Network on CT Brain Hemorrhage Image Segmentation. <http://citeseer.ist.psu.edu/504323.html>

[11] H. C. C. Tan and L. C. De Silva. Human Activity Recognition by Head Movement using Elman Network and Neuro-Markovian Hybrids. ICVNZ'03  
[http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ\\_58.pdf](http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ_58.pdf)

[12] Ling Li, Zhidong Deng, and Bo Zhang. A Fuzzy Elman Neural Network.  
<http://citeseer.ist.psu.edu/479449.html>

© Graphics & Media lab ([cgm@graphics.cs.msu.su](mailto:cgm@graphics.cs.msu.su))

При использовании материалов в сети Интернет или бумажной прессе ссылка на сайт ([cgm.graphicon.ru](http://cgm.graphicon.ru)) обязательна.